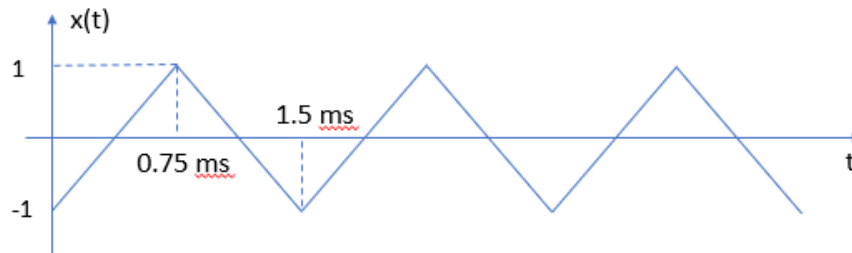


Número

Nome

1. Considere o sinal periódico  $x(t)$  representado na figura.



- a) (2.0 val.) A potência do sinal  $x(t)$  é:

☐ 0 W

☐  $1/3 \times 10^{-3}$  W

☐ 1/2 W

☐ Nenhuma das anteriores

- b) (2.0 val.) Admita que a codificação PCM é realizada com uma frequência de amostragem de 8kHz e por um quantificador MIDRISE com 8 níveis de quantificação. Represente os códigos respeitantes às duas primeiras amostras.

- c) (2.0 val.) Adicionou-se no emissor um código cíclico H(7,4). Tendo recebido a sequência [0101011], pode concluir que se detectou:

☐ 0 bits errados

☐ 1 bits errado

☐ 2 bits errados

☐ Nenhuma das anteriores

- d) (2.0 val.) Para melhorar o BER, em alternativa ao código usado na alínea anterior, a melhor solução é usar:

☐ Código de Hamming H(15,11)

☐ Código de Repetição R(3)

☐ Código de Repetição R(5)

2. Considere um sistema de transmissão Manchester diferencial, com um factor de *roll-off* de 0.50 e um débito binário de 24 kbps. A amplitude máxima do sinal a ser transmitido é de 5V. Sabe-se que o canal de comunicação (AWGN), cujo ruído tem uma densidade espectral de potência de  $\frac{N_0}{2} = 10^{-6}$  W/Hz, tem uma atenuação de 5 dB e a frequência máxima de utilização é de 1 MHz.

- a) (2.0 val.) A potência do sinal recebida no receptor é:

☐ 2.5 W

☐ 5.0 W

☐ 7.9 W

☐ 25 W

- b) (2.0 val.) Considerando que a potência recebida é de 1W, o BER no receptor é inferior a:

☐  $1 \times 10^{-11}$ 
☐  $5 \times 10^{-11}$ 
☐  $7 \times 10^{-11}$ 
☐  $11 \times 10^{-11}$ 

- c) (2.0 val.) Se quiser melhorar o sistema, o melhor código linha para se usar nestas condições é:

☐ UNRZ

☐ Manchester

☐ PNRZ

3. Considere o seguinte código:

```
def My_function(code_array, Eb=1, P=20):
    sinal_total= []
    for x in range(int(len(code_array))):
        bits = code_array[x]
        sinal= []
        if(bits==1):
            for t in range(P):
                sinal= np.append(sinal, np.sqrt((2*Eb)/P)*np.sin(2* np.pi /P * t ) )
        else:
            for t in range(P):
                sinal= np.append(sinal, np.sqrt((2*Eb)/P)*np.sin(2* np.pi /P * t +np.pi) )
        sinal_total = np.append(sinal_total, sinal)
    return sinal_total
```

a) (2.0 val.) Descreva o que faz o código.

b) (2.0 val.) Optimize o código.

4. (2.0 val.) Descreva a funcionalidade do código seguinte:

```
def fun_question(code, Eb = 1, P = 800):

    A= np.sqrt(2*Eb/P)
    c= np.transpose(np.asmatrix(2.*np.asarray(code)-1))
    t= np.asmatrix(np.arange(0,P))
    fc= 10/P
    fi= 10/2/P
    signal = A*np.cos( 2*np.pi*(fc+c*fi) * t)
    return np.array(signal.flatten())[0]
```

## Soluções

$$1a) P = \frac{1}{T_0} \int_0^{T_0} x^2(t) dt = A^2 T_0 / 3 = 1/3 \times 10^{-3} \text{ W}$$

$$1b) L = 8 \\ R = \log_2 8 = 3$$

Intervalo de quantificação	Valores	Código
1.00		
	0.875	100
0.75		
	0.625	101
0.50		
	0.375	111
0.25		
	0.125	110
0.00		
	-0.125	010
-0.25		
	-0.375	011
-0.50		
	-0.625	001
-0.75		
	-0.875	000
-1.00		

$$x(0) = -1 \rightarrow x_q(0) = -0.875 \rightarrow 000 \\ x(1/8000) = -0.66 \rightarrow x_q(1/8000) = -0.625 \rightarrow 001$$

1c) Deteta-se um bit errado

$$\begin{array}{r} 0101011 \quad | \quad 1011 \\ \underline{1011} \\ 111 \end{array}$$

1d) R(5) melhora o BER

$$2a) E_b = A^2 T_b \\ S_T = E_b R_b = A^2 = 25 \text{ W} \\ S_R = \frac{S_T}{10^{\alpha \tau_n / 10}} = 25 / \sqrt{(10)} = 7.9 \text{ W}$$

$$2b) E_b = S_R / R_b = \frac{1}{24 \times 10^3} = 41.66 \times 10^{-6} \\ BER = \operatorname{erfc} \sqrt{\frac{E_b}{N_o}} = \operatorname{erfc}(4.56) = 1.08 \times 10^{-10} < 11 \times 10^{-11}$$

2c) PNRZ. PNRZ e Manchester ambos melhoram o BER comparativamente com o Manchester diferencial, mas o PNRZ ocupa menos Largura de banda. O UNRZ tem pior BER.

3a) Realiza uma modulação BPSK

3b) -

4. Realiza uma modulação BFSK